CS-4341/6341: Project Review Form

***Instructions**:Fill out this form before class and then one person in your team emails its reviews to the instructor (as PDF attachments called review1.pdf and review2.pdf) . You will fill out one form per review. Do not share this review with any other team. Your review is confidential – you will not be identifed as reviewers to the team whose project you are reviewing.*

Your (review) team letter: P
Name of zip file (of the project you are reviewing):

Haiwen-Yihao.zip

1. Describe in a few sentences (your own words) the main idea in this project.

   The Markov Chains is a mathematical model that process the transitions with some rules. There are two properties need to be test in this project. First is the limit of state when the process last for a long time. Second is detailed balance equation.

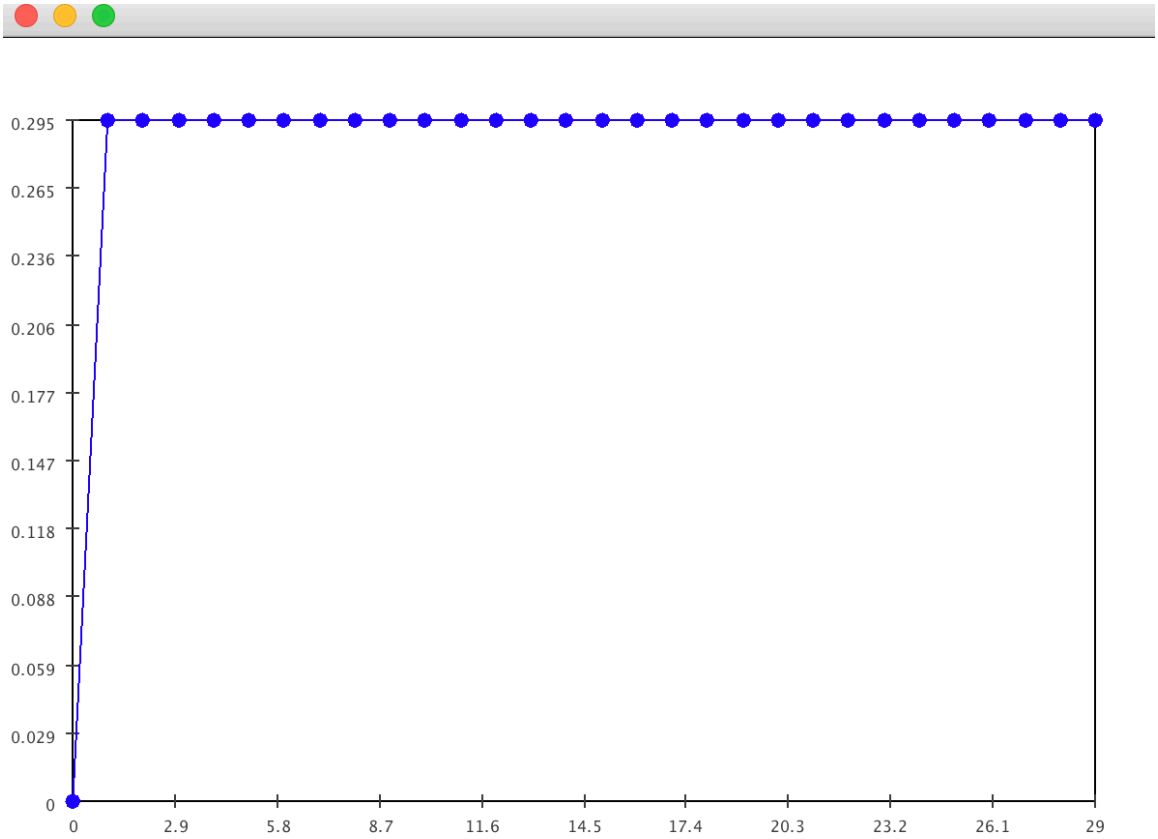
2. Were you able to get the demo working? If not, describe the issues faced.
   Yes.

3. Rate from 1 to 5 (1=weak, 5=strong) the following:

   a. The quality of description of the main idea: 4
   b. The ease with which the demo compiled and ran: 5
   c. The extent to which the demo aligned with the main idea: 3
   d. The quality of description of the exercise: 3
   e. The appropriateness of the exercise to the main idea: 4
   f. The extent to which you learned by doing the exercise: 4
   g. Average rating (average of ratings in a through f): : 4

4. In the space below, write your solution to the exercise, and bring along code (to class) to demonstrate completion of the exercise. If you could not complete the exercise, explain why. Use additional pages if needed.

   Lack of some necessary directions about how the equation works and lose some file like *Function.java* and *SimplePlotPanel.java.*



f1

```
for (int k = 0; k < num; k++) {
        // Calculate the next state using previous state and transition matrix
        if( cnt == 0){
           f1.add(k, next[0]);
           f1.add(k, next[1]);
           f1.add(k, next[2]);
        }

         for (int i = 0; i < 3; i++) {
           next[i] = 0;
        }

        for (int i = 0; i < 3; i++) {
           for (int j = 0; j < 3; j++) {
              next[i] += prev[j] * transition[j][i];
           }
        }
           f1.add(k, next[0]);
           f1.add(k, next[1]);
           f1.add(k, next[2]);
     }

  }
```